Numerical Methods in Economics

Alexis Akira Toda*

November 15, 2017

1 Polynomial interpolation

Polynomials are great because we can differentiate and integrate them very easily. Since a degree $n-1$ polynomial is determined by n coefficients, once we specify n points on the xy plane, there exists (at most) one polynomial that passes through these points.

1.1 Lagrange interpolation

The Lagrange interpolation gives an explicit formula.

Proposition 1. *Let $x_1 < \cdots < x_n$ and define the k-th Lagrange polynomial*

$$L_k(x) = \frac{\prod_{l \neq k}(x - x_l)}{\prod_{l \neq k}(x_k - x_l)}$$

for $k = 1, \ldots, n$. Then

$$p(x) = \sum_{k=1}^{n} y_k L_k(x)$$

is the unique polynomial of degree up to $n-1$ satisfying $p(x_k) = y_k$ for $k = 1, \ldots, n$.

Proof. By the definition of $L_k(x)$, we have $L_k(x_l) = \delta_{kl}$ (Kronecker's delta[1]). Therefore for all l, we have

$$p(x_l) = \sum_{k=1}^{n} y_k L_k(x_l) = \sum_{k=1}^{n} y_k \delta_{kl} = y_l.$$

Clearly $L_k(x)$ is a polynomial of degree $n-1$, so $p(x)$ is a polynomial of degree up to $n-1$. □

If we interpolate a function $f(x)$ at the points $x_1 < \cdots < x_n$ by a degree $n-1$ polynomial, what is the error? The following proposition gives an error bound if f is sufficiently smooth.

*atoda@ucsd.edu

[1] https://en.wikipedia.org/wiki/Kronecker_delta

Proposition 2. *Let f be C^n and p_{n-1} be the interpolating polynomial of f at $x_1, \ldots, x_n$. Then for any x, there exists ξ in the convex hull of $\{x, x_1, \ldots, x_n\}$ such that*

$$f(x) - p_{n-1}(x) = \frac{f^{(n)}(\xi)}{n!} \prod_{k=1}^{n} (x - x_k).$$

Proof. Let $I = \operatorname{co}\{x, x_1, \ldots, x_n\}$. For any $t \in I$, let $R(t) = f(t) - p_{n-1}(t)$ be the error term and define

$$g(t) = R(t)S(x) - R(x)S(t),$$

where $S(t) = \prod_{k=1}^{n}(t - x_k)$. Clearly $g(x) = 0$. Furthermore, since $R(x_k) = S(x_k) = 0$, we have $g(x_k) = 0$ for $k = 1, \ldots, n$. In general, if g is differentiable and $g(a) = g(b) = 0$, then there exists $c \in (a, b)$ such that $g'(c) = 0$ (Rolle's theorem[2]). Therefore there exist n points $y_1, \ldots, y_n$ between $x, x_1, \ldots, x_n$ such that $g'(y_k) = 0$ for $k = 1, \ldots, n$. Continuing this argument, there exists $\xi \in I$ such that $g^{(n)}(\xi) = 0$. But since S is a degree n polynomial with leading coefficient 1, we have $S^{(n)} = n!$, so

$$0 = g^{(n)}(\xi) = R^{(n)}(\xi)S(x) - R(x)n!.$$

Since $R(t) = f(t) - p_{n-1}(t)$ and $\deg p_{n-1} \le n - 1$, we obtain $R^{(n)}(\xi) = f^{(n)}(\xi)$. Therefore

$$f(x) - p_{n-1}(x) = R(x) = \frac{1}{n!} f^{(n)}(\xi) S(x) = \frac{f^{(n)}(\xi)}{n!} \prod_{k=1}^{n} (x - x_k). \quad \square$$

1.2 Chebyshev polynomials

If we want to interpolate a function on an interval by a polynomial, how should we choose the interpolation nodes $x_1, \ldots, x_n$? First, without loss of generality we may assume that the interval is $[-1, 1]$. Since $f^{(n)}(\xi)$ depends on the particular function f but $\prod_{k=1}^{n}(x - x_k)$ does not, it makes sense to find $x_1, \ldots, x_n$ so as to minimize

$$\max_{x \in [-1,1]} \left| \prod_{k=1}^{n} (x - x_k) \right|.$$

Chebyshev[3] has solved this problem a long time ago.

The degree n Chebyshev polynomial $T_n(x)$ is obtained by expanding $\cos n\theta$ as a degree n polynomial of $\cos\theta$ and setting $x = \cos\theta$. For instance,

$$\begin{aligned}
\cos 0\theta &= 1 && \implies T_0(x) = 1, \\
\cos\theta &= \cos\theta && \implies T_1(x) = x, \\
\cos 2\theta &= 2\cos^2\theta - 1 && \implies T_2(x) = 2x^2 - 1,
\end{aligned}$$

and so on. In general, adding

$$\begin{aligned}
\cos(n+1)\theta &= \cos n\theta \cos\theta - \sin n\theta \sin\theta, \\
\cos(n-1)\theta &= \cos n\theta \cos\theta + \sin n\theta \sin\theta,
\end{aligned}$$

[2] `https://en.wikipedia.org/wiki/Rolle's_theorem`
[3] `https://en.wikipedia.org/wiki/Pafnuty_Chebyshev`

and setting $x = \cos n\theta$, we obtain

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

Theorem 3. *The solution of*

$$\min_{x_1,\ldots,x_n} \max_{x\in[-1,1]} \left| \prod_{k=1}^{n} (x - x_k) \right|$$

is given by $x_k = \cos \frac{2k-1}{2n}\pi$*, in which case* $\prod_{k=1}^{n}(x - x_k) = 2^{1-n}T_n(x)$.

Proof. Let $p(x) = 2^{1-n}T_n(x)$. By the above recursive formula, the leading coefficient of $T_n(x)$ is 2^{n-1}. Therefore the leading coefficient of $p(x)$ is 1. Since $p(\cos\theta) = 2^{1-n}\cos n\theta$, clearly $\sup_{x\in[-1,1]} |p(x)| = 2^{1-n}$. Suppose that there exists a degree n polynomial $q(x)$ with leading coefficient 1 such that $\sup_{x\in[-1,1]} |q(x)| < 2^{1-n}$. Again since $p(\cos\theta) = 2^{1-n}\cos n\theta$, we have $p(x) = (-1)^k 2^{1-n}$ at $x = y_k = \cos k\pi/n$, where $k = 0, 1, \ldots, n$. Since $|q(x)| < 2^{1-n}$ for all $x \in [-1, 1]$, by the intermediate value theorem there exits $z_1, \ldots, z_n$ between $y_0, \ldots, y_n$ such that $p(z_k) - q(z_k) = 0$. But since p, q are polynomials of degree n with leading coefficient 1, $r(x) := p(x) - q(x)$ is a polynomial of degree up to $n - 1$. Since $r(z_k) = 0$ for $k = 1, \ldots, n$, it must be $r(x) \equiv 0$ or $p \equiv q$, which is a contradiction. Therefore $\prod_{k=1}^{n}(x - x_k) = 2^{1-n}T_n(x)$, so $x_k = \cos \frac{2k-1}{2n}\pi$ for $k = 1, \ldots, n$. □

2 Quadrature

Many economic problems involve maximizing the expected value. For example, a typical optimal portfolio problem looks like

$$\max_{\theta} \mathrm{E}[u(R(\theta)w)],$$

where w is initial wealth, u is a Bernoulli utility function, and $R(\theta)$ denotes the portfolio return. Since expectations are integrals, and many integrals cannot be computed explicitly, we need methods to numerically evaluate the integrals, which are called *quadrature* (or numerical integration).

A typical quadrature looks like

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{n=1}^{N} w_n f(x_n),$$

where f is a general integrand and $\{x_n\}_{n=1}^N$ are nodes and $\{w_n\}_{n=1}^N$ are weights of the quadrature rule. See Davis and Rabinowitz (1984) for a standard textbook treatment.

2.1 Newton-Cotes quadrature

The simplest quadrature rule is to divide the interval $[a, b]$ into $N - 1$ even-spaced subintervals (so $x_n = a + \frac{n-1}{N-1}(b - a)$ for $n = 1, \ldots, N$) and choose the weights $\{w_n\}_{n=1}^N$ so that one can integrate all polynomials of degree $N - 1$ or

less exactly. This quadrature rule is known as the N-point Newton-Cotes rule. Since we can map the interval $[0,1]$ to $[a,b]$ through the linear transformation $x \mapsto a + (b-a)x$, without loss of generality let us assume $a = 0$ and $b = 1$. Let us consider several cases.

2.1.1 $N = 2$ (trapezoidal rule)

The 2-point Newton-Cotes rule is known as the trapezoidal rule. In this case we have $x_n = 0, 1$, and we choose w_1, w_2 to integrate a linear function exactly. Therefore

$$\begin{aligned} 1 &= \int_0^1 1 \,\mathrm{d}x = w_1 + w_2, \\ \frac{1}{2} &= \int_0^1 x \,\mathrm{d}x = w_2, \end{aligned}$$

so solving these equations we obtain $w_1 = w_2 = \frac{1}{2}$. Therefore

$$\int_a^b f(x) \,\mathrm{d}x \approx \frac{b-a}{2}(f(a) + f(b)).$$

Let us estimate the error of this approximation. Let $p(x)$ be the degree 1 interpolating polynomial of f at $x = a, b$. Since p agrees with f at a, b, clearly

$$\int_a^b p(x) \,\mathrm{d}x = \frac{b-a}{2}(f(a) + f(b)).$$

Therefore by Proposition 2, we obtain

$$\begin{aligned} \int_a^b f(x) \,\mathrm{d}x - \frac{b-a}{2}(f(a) + f(b)) &= \int_a^b (f(x) - p(x)) \,\mathrm{d}x \\ &= \int_a^b \frac{f''(\xi(x))}{2}(x-a)(x-b) \,\mathrm{d}x, \end{aligned}$$

where $\xi(x) \in (a,b)$. Since $(x-a)(x-b) < 0$ on (a,b), by the mean value theorem for Rieman-Stieltjes integrals, there exists $c \in (a,b)$ such that

$$\int_a^b \frac{f''(\xi(x))}{2}(x-a)(x-b) \,\mathrm{d}x = \frac{f''(c)}{2} \int_a^b (x-a)(x-b) \,\mathrm{d}x = -\frac{f''(c)}{12}(b-a)^3.$$

Therefore we can estimate the error as

$$\left| \int_a^b f(x) \,\mathrm{d}x - \frac{b-a}{2}(f(a) + f(b)) \right| \leq \frac{\|f''\|}{12}(b-a)^3.$$

2.1.2 $N = 3$ (Simpson's rule)

The 3-point Newton-Cotes rule is known as Simpson's rule. In this case we have $x_n = 0, 1/2, 1$, and we choose w_1, w_2, w_3 to integrate a quadratic function

exactly. Therefore

$$\begin{aligned}1 &= \int_0^1 1\,\mathrm{d}x = w_1 + w_2 + w_3,\\ \frac{1}{2} &= \int_0^1 x\,\mathrm{d}x = \frac{1}{2}w_2 + w_3,\\ \frac{1}{3} &= \int_0^1 x^2\,\mathrm{d}x = \frac{1}{4}w_2 + w_3,\end{aligned}$$

so solving these equations we obtain $w_1 = w_3 = \frac{1}{6}$ and $w_2 = \frac{2}{3}$. Therefore

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right).$$

Interestingly, since

$$\frac{1}{4} = \int_0^1 x^3\,\mathrm{d}x = \frac{1}{8}w_2 + w_3,$$

Simpson's rule actually integrates polynomials of degree 3 exactly (even though it is not designed to do so).

To estimate the error of Simpson's rule, take any point $d \in (a,b)$ and let $p(x)$ be a degree 3 interpolating polynomial of f at $x = a, \frac{a+b}{2}, b, d$. Since Simpson's rule integrates degree 3 polynomials exactly, by Proposition 2 we have

$$\begin{aligned}&\int_a^b f(x)\,\mathrm{d}x - \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)\\ &= \int_a^b (f(x) - p(x))\,\mathrm{d}x = \int_a^b \frac{f^{(4)}(\xi(x))}{4!}(x-a)\left(x - \frac{a+b}{2}\right)(x-b)(x-d)\,\mathrm{d}x.\end{aligned}$$

Since $d \in (a,b)$ is arbitrary, we can take $d = \frac{a+b}{2}$. Since

$$(x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b) < 0$$

on (a,b) almost everywhere, as before we can apply the mean value theorem. Using the change of variable $x = \frac{a+b}{2} + \frac{b-a}{2}t$, we can compute

$$\begin{aligned}&\int_a^b (x-a)\left(x - \frac{a+b}{2}\right)^2 (x-b)\,\mathrm{d}x\\ &= \left(\frac{b-a}{2}\right)^5 \int_{-1}^1 (t+1)t^2(t-1)\,\mathrm{d}t = -\frac{1}{120}(b-a)^5.\end{aligned}$$

Since $4! = 24$ and $24 \times 120 = 2880$, the integration error is

$$\left|\int_a^b f(x)\,\mathrm{d}x - \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)\right| \le \frac{\left\|f^{(4)}\right\|}{2880}(b-a)^5.$$

2.2 Compound rule

Newton-Cotes rule with $N \geq 4$ are almost never used because beyond some order some of the weights $\{w_n\}_{n=1}^N$ become negative, which introduces rounding errors. One way to avoid this problem is to divide the interval $[a, b]$ into N even-spaced subintervals and apply the trapezoidal rule or the Simpson's rule to each subinterval. This method is known as the compound (or composite) rule.

If you use the trapezoidal rule, then there are $N+1$ points. Letting $x_n = n/N$ for $n = 0, 1, \ldots, N$, the formula for $[0, 1]$ is

$$\begin{aligned}\int_0^1 f(x)\,\mathrm{d}x &\approx \sum_{n=1}^{N} \frac{1}{2N}(f(x_{n-1}) + f(x_n)) \\ &= \frac{1}{2N}\left(f(x_0) + 2f(x_1) + \cdots + 2f(x_{N-1}) + f(x_N)\right).\end{aligned}$$

(Just remember that the relative weights are 1 at endpoints and 2 in between.) Since $b - a = 1/N$ and there are N subintervals, the error of the $(N+1)$-point trapezoidal rule is $\frac{\|f''\|}{12}N^{-2}$.

If you use Simpson's rule, then there are 3 points on each subinterval, of which there are N, and $N-1$ endpoints are counted twice. Therefore the total number of points is $3N - (N-1) = 2N + 1$. Letting $x_n = n/(2N)$ for $n = 0, 1, \ldots, N$, the formula for $[0, 1]$ is

$$\begin{aligned}\int_0^1 f(x)\,\mathrm{d}x &\approx \sum_{n=1}^{N} \frac{1}{6N}(f(x_{2n-2}) + 4f(x_{2n-1}) + f(x_{2n})) \\ &= \frac{1}{6N}\left(f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 4f(x_{2N-1}) + f(x_{2N})\right).\end{aligned}$$

(Just remember that the relative weights are 1 at endpoints, and they alternate like $4, 2, 4, 2, \ldots, 4, 2, 4$ in between.) Since $b - a = 1/N$ and there are N subintervals, the error of the $(2N+1)$-point Simpson's rule is $\frac{\|f^{(4)}\|}{2880}N^{-4}$.

Since the quadrature weights are given explicitly for trapezoidal and Simpson's rule, it is straightforward to write programs that compute numerical integrals. The tables below show the $\log_{10}$ relative errors of integrals over the interval $[0, 1]$ ($\log_{10}\left|\widehat{I}/I - 1\right|$, where I is the true integral and $\widehat{I}$ is the numerical one) for several functions when we use the N-point compound trapezoidal and Simpson's rule. As the above error analysis suggests, errors tend to be smaller when the integrand is smoother (has higher order derivatives). Furthermore, Simpson's rule is more accurate than the trapezoidal rule.

Table 1. $\log_{10}$ relative errors of compound trapezoidal rule.

# points	$x^{1/2}$	$x^{3/2}$	$x^{5/2}$	$x^{7/2}$	$x^{9/2}$	e^x
3	-1.0238	-1.1743	-0.7343	-0.4896	-0.3041	-1.6830
5	-1.4550	-1.7558	-1.3394	-1.0875	-0.8937	-2.2838
9	-1.8926	-2.3438	-1.9427	-1.6885	-1.4928	-2.8855
17	-2.3346	-2.9361	-2.5452	-2.2902	-2.0941	-3.4874
33	-2.7795	-3.5314	-3.1474	-2.8922	-2.6960	-4.0895
65	-3.2264	-4.1287	-3.7495	-3.4943	-3.2980	-4.6915

Table 2. $\log_{10}$ relative errors of compound Simpson's rule.

# points	$x^{1/2}$	$x^{3/2}$	$x^{5/2}$	$x^{7/2}$	$x^{9/2}$	e^x
3	-1.3676	-2.2275	-2.3780	-1.8192	-1.1040	-3.4722
5	-1.8179	-2.9667	-3.3705	-2.9823	-2.3199	-4.6667
9	-2.2691	-3.7142	-4.3841	-4.1584	-3.5289	-5.8684
17	-2.7206	-4.4649	-5.4112	-5.3435	-4.7350	-7.0720
33	-3.1722	-5.2168	-6.4470	-6.5346	-5.9399	-8.2759
65	-3.6237	-5.9692	-7.4884	-7.7297	-7.1443	-9.4800

2.3 Gaussian quadrature

In the Newton-Cotes quadrature, we assume that the nodes are even-spaced, but of course this is not necessary. Can we do better by choosing the quadrature nodes optimally? In general, consider the integral

$$\int_a^b w(x)f(x)\,\mathrm{d}x,$$

where $-\infty \le a < b \le \infty$ are endpoints of integration, $w(x) > 0$ is some (fixed) weighting function, and f is a general integrand. A typical example is $a = -\infty$, $b = \infty$, and $w(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-(x-\mu)^2/2\sigma^2}$, in which case we want to compute the expectation $\mathrm{E}[f(X)]$ when $X \sim N(\mu, \sigma^2)$.

In the discussion below, let us omit a, b (so $\int$ means $\int_a^b$) and assume that $\int w(x)x^n\,\mathrm{d}x$ exists for all $n \ge 0$. For functions f, g, let us define the inner product (f, g) by

$$(f, g) = \int_a^b w(x)f(x)g(x)\,\mathrm{d}x.$$

Let $p_n^*(x)$ be the orthogonal polynomial of degree n corresponding to the weighting function $w(x)$. This means that $(p_m^*, p_n^*) = \delta_{mn}$, where δ_{mn} is Kronecker's delta. Orthogonal polynomials (with positive leading coefficients) uniquely exist and can be constructed using the Gram-Schmidt orthonormalization.[4] In fact, we can recursively compute the orthogonal polynomials using the three-term recurrence relation (TTRR) as follows.

Proposition 4 (TTRR)**.** *Let $p_{-1}^*(x) = 0$, $p_0(x) = 1$, and for $n \ge 0$ define*

$$\begin{aligned} p_n^*(x) &= p_n(x)/(p_n, p_n)^{1/2}, \\ p_{n+1}(x) &= xp_n^*(x) - (xp_n^*, p_n^*)p_n^*(x) - (p_n, p_n)^{1/2}p_{n-1}^*(x). \end{aligned}$$

Then $p_n^*(x)$ is the n-degree orthogonal polynomial.

Proof. Clearly $(p_n^*, p_n^*) = 1$ for all n. Therefore it suffices to show that $(p_m^*, p_n^*) = 0$ whenever $m < n$. Let us prove this by induction on $n \ge 0$. If $n = 0$, there is nothing to prove. If $n = 1$, since

$$p_1(x) = xp_0^*(x) - (xp_0^*, p_0^*)p_0^*(x),$$

taking the inner product with $p_0^*(x)$, we obtain

$$(p_1, p_0^*) = (xp_0^*, p_0^*) - (xp_0^*, p_0^*) = 0.$$

[4] `https://en.wikipedia.org/wiki/Gram-Schmidt_process`

Therefore by rescaling p_1 to p_1^*, we obtain $(p_0^*, p_1^*) = 0$. Suppose the claim holds up to n. Then for $n+1$, since

$$p_{n+1}(x) = xp_n^*(x) - (xp_n^*, p_n^*)p_n^*(x) - (p_n, p_n)^{1/2}p_{n-1}^*(x),$$

taking the inner product with p_n^*, we obtain

$$(p_{n+1}, p_n^*) = (xp_n^*, p_n^*) - (xp_n^*, p_n^*) = 0.$$

Similarly, taking the inner product with p_{n-1}^*, we obtain

$$(p_{n+1}, p_{n-1}^*) = (xp_n^*, p_{n-1}^*) - (p_n, p_n)^{1/2} = (p_n^*, xp_{n-1}^*) - (p_n, p_n)^{1/2}.$$

Let $k_n > 0$ be the leading coefficient of p_n^*. By the recursive definition and the normalization, we have

$$k_n = k_{n-1}/(p_n, p_n)^{1/2}.$$

Since xp_{n-1}^* is an n-degree polynomial with leading coefficient k_{n-1}, it can be expanded as

$$xp_{n-1}^*(x) = \frac{k_{n-1}}{k_n}p_n^*(x) + \text{low order polynomials}.$$

Therefore

$$(p_{n+1}, p_{n-1}^*) = \frac{k_{n-1}}{k_n} - (p_n, p_n)^{1/2} = 0. \quad \square$$

The following lemma shows that an n-degree orthogonal polynomial has exactly n real roots (so they are all simple).

Lemma 5. *$p_n^*(x)$ has exactly n real roots on (a, b).*

Proof. By the fundamental theorem of algebra, $p_n^*(x)$ has exactly n roots in $\mathbb{C}$. Suppose on the contrary that $p_n^*(x)$ has less than n real roots on (a, b). Let $x_1, \ldots, x_k$ $(k < n)$ those roots at which $p_n(x)$ changes its sign. Let $p(x) = (x - x_1) \cdots (x - x_k)$. Since $p_n^*(x)p(x) > 0$ (or < 0) almost everywhere on (a, b), we have

$$(p_n^*, p) = \int w(x)p_n^*(x)p(x)\,\mathrm{d}x \neq 0.$$

On the other hand, since $\deg p = k < n$ and orthogonal polynomials are linearly independent, we can express p as a weighted sum of p_m^*'s, where $m \leq k$. By the definition of the orthogonal polynomials, it follows that $(p_n^*, p) = 0$, which is a contradiction. $\square$

The following theorem shows that using the n roots of the degree n orthogonal polynomial as quadrature nodes and choosing specific weights, we can integrate all polynomials of degree up to $2n - 1$ exactly. This is known as Gaussian quadrature.

Theorem 6 (Gaussian quadrature)**.** *Let $a < x_1 < \cdots < x_n < b$ be the n roots of p_n^* and define*

$$w_k = \int w(x)L_k(x)\,\mathrm{d}x$$

for $k = 1, \ldots, n$, where $L_k(x)$ is as in Proposition 1. Then

$$\int w(x)p(x)\,\mathrm{d}x = \sum_{k=1}^{n} w_k p(x_k)$$

for all polynomials $p(x)$ of degree up to $2n-1$.

Proof. Since $\deg p \le 2n-1$ and $\deg p_n^* = n$, we can write

$$p(x) = p_n^*(x)q(x) + r(x),$$

where $\deg q, \deg r \le n-1$. Since q can be expressed as a linear combination of orthogonal polynomials of degree up to $n-1$, we have $(p_n^*, q) = 0$. Hence

$$\int w(x)p(x)\,\mathrm{d}x = (p_n^*, q) + \int w(x)r(x)\,\mathrm{d}x = \int w(x)r(x)\,\mathrm{d}x.$$

On the other hand, since $\{x_k\}_{k=1}^n$ are roots of p_n^*, we have

$$p(x_k) = p_n^*(x_k)q(x_k) + r(x_k) = r(x_k)$$

for all k, so in particular

$$\sum_{k=1}^{n} w_k p(x_k) = \sum_{k=1}^{n} w_k r(x_k).$$

Therefore it suffices to show the claim for polynomials r of degree up to $n-1$. Since by Proposition 1 any such polynomial can be represented as a linear combination of L_k's, it suffices to show the claim for all L_k's. But since by definition

$$\int w(x)L_k(x)\,\mathrm{d}x = w_k = \sum_{l=1}^{n} w_l L_k(x_l),$$

the claim is true. □

In practice, how can we compute the nodes $\{x_n\}_{n=1}^N$ and weights $\{w_n\}_{n=1}^N$ of the N-point Gaussian quadrature? The solution is given by the following Golub-Welsch algorithm.

Theorem 7 (Golub-Welsch)**.** *Let $k_n > 0$ be the leading coefficient of p_n^*, $\alpha_n = k_{n-1}/k_n > 0$, and $\beta_n = (xp_n^*, p_n^*)$. Define the $N \times N$ symmetric matrix J_N by*

$$J_N = \begin{bmatrix} \beta_0 & \alpha_1 & 0 & \cdots & 0 \\ \alpha_1 & \beta_1 & \alpha_2 & \ddots & \vdots \\ 0 & \alpha_2 & \beta_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha_{N-1} \\ 0 & \cdots & 0 & \alpha_{N-1} & \beta_{N-1} \end{bmatrix}.$$

Then the Gaussian quadrature nodes $\{x_n\}_{n=1}^N$ are eigenvalues of J_N, and the weights $\{w_n\}_{n=1}^N$ are given by

$$\frac{1}{w_n} = \sum_{k=0}^{N-1} p_k^*(x_n)^2 > 0.$$

Proof. By the proof of Proposition 4, we have $(p_n, p_n)^{1/2} = k_{n-1}/k_n = \alpha_n$. Therefore TTRR becomes

$$p_{n+1}(x) = xp_n^*(x) - \beta_n p_n^*(x) - \alpha_n p_{n-1}^*(x).$$

Since $p_{n+1}^*(x) = p_{n+1}(x)/(p_{n+1}, p_{n+1})^{1/2} = p_{n+1}(x)/\alpha_{n+1}$, TTRR becomes

$$\alpha_n p_{n-1}^*(x) + \beta_n p_n^*(x) + \alpha_{n+1} p_{n+1}^*(x) = xp_n^*(x).$$

In particular, setting $x = x_k$ (where x_k is a root of p_N^*), we obtain

$$\alpha_n p_{n-1}^*(x_k) + \beta_n p_n^*(x_k) + \alpha_{n+1} p_{n+1}^*(x_k) = xp_n^*(x_k)$$

for all n and $k = 1, \dots, N$. Since $p_{-1}^* = 0$ by definition and $p_N^*(x_k) = 0$, letting $P(x) = (p_0^*(x), \dots, p_{N-1}^*(x))'$ and collecting the above equation into a vector we obtain

$$J_N P(x_k) = x_k P(x_k)$$

for $k = 1, \dots, N$. Define the $N \times N$ matrix P by $P = (P(x_1), \dots, P(x_N))$. Since

$$\delta_{mn} = (p_m^*, p_n^*) = \int w(x) p_m^*(x) p_n^*(x)\, \mathrm{d}x = \sum_{k=1}^{N} w_k p_m^*(x_k) p_n^*(x_k)$$

for $m, n \le N-1$ because the Gaussian quadrature integrates all polynomials of degree up to $2n-1$ exactly, letting $W = \operatorname{diag}(w_1, \dots, w_N)$ we have

$$PWP' = I.$$

Therefore P, W are invertible. Solving for W and taking the inverse, we obtain

$$W^{-1} = P'P \iff \frac{1}{w_n} = \sum_{k=0}^{N-1} p_k^*(x_n)^2 > 0$$

for all n. Since $J_N P(x_n) = x_n P(x_n)$ for $n = 1, \dots, N$, collecting into a matrix we obtain

$$J_N P = XP \iff P^{-1} J_N P = X,$$

where $X = \operatorname{diag}(x_1, \dots, x_N)$. Therefore $\{x_n\}_{n=1}^N$ are eigenvalues of J_N. □

Below are some examples. By googling you can find subroutines in Matlab or whatever language that compute the nodes and weights of these quadratures.

Example 1. The case $(a, b) = (-1, 1)$, $w(x) = 1$ is known as the Gauss-Legendre quadrature.

Example 2. The case $(a, b) = (-1, 1)$, $w(x) = 1/\sqrt{1-x^2}$ is known as the Gauss-Chebyshev quadrature. It is useful for computing Fourier coefficients (through the change of variable $x = \cos\theta$).

Example 3. The case $(a, b) = (-\infty, \infty)$, $w(x) = \mathrm{e}^{-x^2}$ is known as the Gauss-Hermite quadrature, which is useful for computing the expectation with respect to the normal distribution.

Example 4. The case $(a, b) = (0, \infty)$, $w(x) = \mathrm{e}^{-x}$ is known as the Gauss-Laguerre quadrature, which is useful for computing the expectation with respect to the exponential distribution.

The table below shows the $\log_{10}$ relative errors when using the N-point Gauss-Legendre quadrature. You can see that Gaussian quadrature is overwhelmingly more accurate than Newton-Cotes.

Table 3. $\log_{10}$ relative errors of compound trapezoidal rule.

# points	$x^{1/2}$	$x^{3/2}$	$x^{5/2}$	$x^{7/2}$	$x^{9/2}$	e^x
3	-2.4237	-3.3289	-3.8570	-4.0525	-3.8824	-6.3191
5	-3.0245	-4.3578	-5.3560	-6.0948	-6.6082	-12.4194
9	-3.7418	-5.5649	-7.0688	-8.3362	-9.4106	-15.9546
17	-4.5396	-6.8986	-8.9436	-10.7592	-12.3913	-15.9546
33	-5.3862	-8.3108	-10.9229	-13.3092	-15.3525	$-\infty$

3 Discretization

If the goal is to solve a single optimization problem that involves expectations (*e.g.*, static optimal portfolio problem), a highly accurate Gaussian quadrature is a natural choice. However, many economic problems are dynamic, in which case one needs to compute conditional expectations. Furthermore, to reduce the computational complexity of the problem, it is desirable that the quadrature nodes are preassigned instead of being dependent on the particular state of the model. Discretization is a useful tool for solving such problems.

3.1 Earlier methods

For concreteness, consider the Gaussian AR(1) process

$$x_t = \rho x_{t-1} + \varepsilon_t, \quad \epsilon_t \sim N(0, \sigma^2).$$

Then the conditional distribution of x_t given x_{t-1} is $N(\rho x_{t-1}, \sigma^2)$. How can we discretize (find a finite-state Markov chain approximation) of this stochastic process?

A classic method is Tauchen (1986) but you should never use it because it is not accurate (so I won't explain further). Similarly, the quantile method in Adda and Cooper (2003) is poor. For Gaussian AR(1) process, the Rouwenhorst (1995) is good because the conditional moments are exact up to order 2 and the method is constructive (does not involve optimization). It is especially useful when $\rho \geq 0.99$.

The Tauchen and Hussey (1991) method, which I explain now, uses the Gaussian quadrature. First consider discretizing $N(0, \sigma^2)$. Letting $\{x_n\}_{n=1}^N$ and $\{w_n\}_{n=1}^N$ be the nodes and weights of the N-point Gauss-Hermite quadrature, since for any integrand g we have

$$\begin{aligned}
\mathrm{E}[g(X)] &= \int_{-\infty}^{\infty} g(x) \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \,\mathrm{d}x \\
&= \int_{-\infty}^{\infty} g(\sqrt{2}\sigma y) \frac{1}{\sqrt{\pi}} \mathrm{e}^{-y^2} \,\mathrm{d}y \\
&\approx \sum_{n=1}^{N} \frac{w_n}{\sqrt{\pi}} g(\sqrt{2}\sigma x_n),
\end{aligned}$$

we can use the nodes $x'_n = \sqrt{2}\sigma x_n$ and weights $w'_n = w_n/\sqrt{\pi}$ to discretize $N(0, \sigma^2)$.

The same idea can be used to discretize the Gaussian AR(1) process. Let us fix the nodes $\{x'_n\}_{n=1}^N$ as constructed above. Since for any integrand g, letting

$\mu = \rho x'_m$ we have

$$\begin{aligned}
\mathrm{E}\left[g(x_t) \,|\, x_{t-1} = x'_m\right] &= \int_{-\infty}^{\infty} g(x) \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \,\mathrm{d}x \\
&= \int_{-\infty}^{\infty} g(x) \mathrm{e}^{-\frac{\mu^2 - 2x\mu}{2\sigma^2}} \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}} \,\mathrm{d}x \\
&\approx \sum_{n=1}^{N} w'_n \mathrm{e}^{-\frac{\mu^2 - 2x'_n\mu}{2\sigma^2}} g(x'_n),
\end{aligned}$$

so we can construct the transition probability matrix $P = (p_{mn})$ by

$$p_{mn} \propto w'_n \mathrm{e}^{-\frac{\mu^2 - 2x'_n\mu}{2\sigma^2}},$$

where $\mu = \rho x'_m$ and the constant of proportionality is determined by $\sum_{n=1}^{N} p_{mn} = 1$. The Tauchen-Hussey method is pretty good if $\rho \leq 0.5$, although a drawback is that it assumes Gaussian shocks. Furthermore, the performance deteriorates when ρ becomes larger.

3.2 Farmer-Tanaka-Toda maximum entropy method

Several papers by me and my coauthors (Tanaka and Toda, 2013, 2015; Farmer and Toda, 2017) provide a more accurate and generally applicable discretization method (so it should be the first choice!). Below I briefly explain the method, but see Farmer and Toda (2017) for more details.

3.2.1 Discretizing probability distributions

Suppose that we are given a continuous probability density function $f : \mathbb{R}^K \to \mathbb{R}$, which we want to discretize. Let X be a random vector with density f, and $g : \mathbb{R}^K \to \mathbb{R}$ be any bounded continuous function. The first step is to pick a quadrature formula

$$\mathrm{E}[g(X)] = \int_{\mathbb{R}^K} g(x) f(x) \,\mathrm{d}x \approx \sum_{n=1}^{N} w_n g(x_n) f(x_n), \tag{1}$$

where N is the number of integration points, $\{x_n\}_{n=1}^N$, and $w_n > 0$ is the weight on the integration point x_n.

For now, we do not take a stance on the choice of the initial quadrature formula, but take it as given. Given the quadrature formula (1), a coarse but valid discrete approximation of the density f would be to assign probability q_n to the point x_n proportional to $w_n f(x_n)$, so

$$q_n = \frac{w_n f(x_n)}{\sum_{n=1}^{N} w_n f(x_n)}. \tag{2}$$

However, this is not necessarily a good approximation because the moments of the discrete distribution $\{q_n\}$ do not generally match those of f.

Tanaka and Toda (2013) propose exactly matching a finite set of moments by updating the probabilities $\{q_n\}$ in a particular way. Let $T : \mathbb{R}^K \to \mathbb{R}^L$

be a function that defines the moments that we wish to match and let $\bar{T} = \int_{\mathbb{R}^K} T(x) f(x)\,\mathrm{d}x$ be the vector of exact moments. For example, if we want to match the first and second moments in the one dimensional case ($K = 1$), then $T(x) = (x, x^2)'$. Tanaka and Toda (2013) update the probabilities $\{q_n\}$ by solving the optimization problem

$$\begin{aligned} \underset{\{p_n\}}{\text{minimize}} \quad & \sum_{n=1}^{N} p_n \log \frac{p_n}{q_n} \\ \text{subject to} \quad & \sum_{n=1}^{N} p_n T(x_n) = \bar{T},\ \sum_{n=1}^{N} p_n = 1,\ p_n \geq 0. \end{aligned} \tag{P}$$

The objective function in the primal problem (P) is the Kullback and Leibler (1951) information of $\{p_n\}$ relative to $\{q_n\}$, which is also known as the relative entropy. This method matches the given moments exactly while keeping the probabilities $\{p_n\}$ as close to the initial approximation $\{q_n\}$ as possible in the sense of the Kullback-Leibler information. Note that since (P) is a convex minimization problem, the solution (if one exists) is unique.

The optimization problem (P) is a constrained minimization problem with a large number (N) of unknowns ($\{p_n\}$) with $L + 1$ equality constraints and N inequality constraints, which is in general computationally intensive to solve. However, it is well-known that entropy-like minimization problems are computationally tractable by using duality theory (Borwein and Lewis, 1991). Tanaka and Toda (2013) convert the primal problem (P) to the dual problem

$$\max_{\lambda \in \mathbb{R}^L} \left[\lambda' \bar{T} - \log \left(\sum_{n=1}^{N} q_n \mathrm{e}^{\lambda' T(x_n)} \right) \right], \tag{D}$$

which is a *low dimensional* (L unknowns) *unconstrained* concave maximization problem and hence computationally tractable. The following theorem shows how the solutions to the two problems (P) and (D) are related. Below, the symbols "int" and "co" denote the interior and the convex hull of sets.

Theorem 8. *1. The primal problem* (P) *has a solution if and only if* $\bar{T} \in \operatorname{co} T(D_N)$. *If a solution exists, it is unique.*

2. The dual problem (D) *has a solution if and only if* $\bar{T} \in \operatorname{int} \operatorname{co} T(D_N)$. *If a solution exists, it is unique.*

3. If the dual problem (D) *has a (unique) solution* λ_N, *then the (unique) solution to the primal problem* (P) *is given by*

$$p_n = \frac{q_n \mathrm{e}^{\lambda_N' T(x_n)}}{\sum_{n=1}^{N} q_n \mathrm{e}^{\lambda_N' T(x_n)}} = \frac{q_n \mathrm{e}^{\lambda_N' (T(x_n) - \bar{T})}}{\sum_{n=1}^{N} q_n \mathrm{e}^{\lambda_N' (T(x_n) - \bar{T})}}. \tag{3}$$

Theorem 8 provides a practical way to implement the Tanaka-Toda method. After choosing the initial discretization $Q = \{q_n\}$ and the moment defining function T, one can numerically solve the unconstrained optimization problem (D). To this end, we can instead solve

$$\min_{\lambda \in \mathbb{R}^L} \sum_{n=1}^{N} q_n \mathrm{e}^{\lambda' (T(x_n) - \bar{T})} \tag{D$'$}$$

because the objective function in (D′) is a monotonic transformation (-1 times the exponential) of that in (D). Since (D′) is an unconstrained convex minimization problem with a (relatively) small number (L) of unknowns (λ), solving it is computationally simple. Letting $J_N(\lambda)$ be the objective function in (D′), its gradient and Hessian can be analytically computed as

$$\nabla J_N(\lambda) = \sum_{n=1}^{N} q_n \mathrm{e}^{\lambda'(T(x_n)-\bar{T})}(T(x_n)-\bar{T}), \tag{4a}$$

$$\nabla^2 J_N(\lambda) = \sum_{n=1}^{N} q_n \mathrm{e}^{\lambda'(T(x_n)-\bar{T})}(T(x_n)-\bar{T})(T(x_n)-\bar{T})', \tag{4b}$$

respectively. In practice, we can quickly solve (D′) numerically using optimization routines by supplying the analytical gradient and Hessian.[5]

If a solution to (D′) exists, it is unique, and we can compute the updated discretization $P = \{p_n\}$ by (3). If a solution does not exist, it means that the regularity condition $\bar{T} \in \operatorname{int} \operatorname{co} T(D_N)$ does not hold and we cannot match moments. Then one needs to select a smaller set of moments. Numerically checking whether moments are matched is straightforward: by (3), (D′), and (4a), the error is

$$\sum_{n=1}^{N} p_n T(x_n) - \bar{T} = \frac{\sum_{n=1}^{N} q_n \mathrm{e}^{\lambda_N'(T(x_n)-\bar{T})}(T(x_n)-\bar{T})}{\sum_{n=1}^{N} q_n \mathrm{e}^{\lambda_N'(T(x_n)-\bar{T})}} = \frac{\nabla J_N(\lambda_N)}{J_N(\lambda_N)}. \tag{5}$$

3.2.2 Discretizing general Markov processes

Next we show how to extend the Tanaka-Toda method to the case of time-homogeneous Markov processes.

Consider the time-homogeneous first-order Markov process

$$P(x_t \le x' | x_{t-1} = x) = F(x', x),$$

where x_t is the vector of state variables and $F(\cdot, x)$ is a cumulative distribution function (CDF) that determines the distribution of $x_t = x'$ given $x_{t-1} = x$. The dynamics of any Markov process are completely characterized by its Markov transition kernel. In the case of a discrete state space, this transition kernel is simply a matrix of transition probabilities, where each row corresponds to a conditional distribution. We can discretize the continuous process x by applying the Tanaka-Toda method to each conditional distribution separately.

More concretely, suppose that we have a set of grid points $D_N = \{x_n\}_{n=1}^N$ and an initial coarse approximation $Q = (q_{nn'})$, which is an $N \times N$ probability transition matrix. Suppose we want to match some conditional moments of x, represented by the moment defining function $T(x)$. The exact conditional moments when the current state is $x_{t-1} = x_n$ are

$$\bar{T}_n = \mathrm{E}\left[T(x_t) \,|\, x_n\right] = \int T(x) \,\mathrm{d}F(x, x_n),$$

[5]Since the dual problem (D) is a concave maximization problem, one may also solve it directly. However, according to our experience, solving (D′) is numerically more stable. This is because the objective function in (D) is close to linear when $\|\lambda\|$ is large, so the Hessian is close to singular and not well-behaved. On the other hand, since the objective function in (D′) is the sum of exponential functions, it is well-behaved.

where the integral is over x, fixing x_n. (If these moments do not have explicit expressions, we can use highly accurate quadrature formulas to compute them.) By Theorem 8, we can match these moments exactly by solving the optimization problem

$$\begin{aligned} &\underset{\{p_{nn'}\}_{n'=1}^N}{\text{minimize}} && \sum_{n'=1}^N p_{nn'} \log \frac{p_{nn'}}{q_{nn'}} \\ &\text{subject to} && \sum_{n'=1}^N p_{nn'} T(x_{n'}) = \bar{T}_n, \ \sum_{n'=1}^N p_{nn'} = 1, \ p_{nn'} \geq 0 \end{aligned} \tag{P_n}$$

for each $n = 1, 2, \ldots, N$, or equivalently the dual problem

$$\min_{\lambda \in \mathbb{R}^L} \sum_{n'=1}^N q_{nn'} \mathrm{e}^{\lambda'(T(x_{n'}) - \bar{T}_n)}. \tag{D'_n}$$

(D'_n) has a unique solution if and only if the regularity condition

$$\bar{T}_n \in \operatorname{int} \operatorname{co} T(D_N) \tag{6}$$

holds. We summarize our procedure in Algorithm 1 below.

Algorithm 1 (Discretization of Markov processes)**.**

1. Select a discrete set of points $D_N = \{x_n\}_{n=1}^N$ and an initial approximation $Q = (q_{nn'})$.
2. Select a moment defining function $T(x)$ and corresponding exact conditional moments $\{\bar{T}_n\}_{n=1}^N$. If necessary, approximate the exact conditional moments with a highly accurate numerical integral.
3. For each $n = 1, \ldots, N$, solve minimization problem (D'_n) for λ_n. Check whether moments are matched using formula (5), and if not, select a smaller set of moments. Compute the conditional probabilities corresponding to row n of $P = (p_{nn'})$ using (3).

The resulting discretization of the process is given by the transition probability matrix $P = (p_{nn'})$. Since the dual problem (D'_n) is an unconstrained convex minimization problem with a typically small number of variables, standard Newton type algorithms can be applied. Furthermore, since the probabilities (3) are strictly positive by construction, the transition probability matrix $P = (p_{nn'})$ is a strictly positive matrix, so the resulting Markov chain is stationary and ergodic.

4 Projection

In economics we often need to solve functional equations. For instance, in an income fluctuation problem, we need to characterize the optimal consumption rule $c(w, y)$ given wealth w and income y. The projection method (a standard reference is Judd, 1992) approximates the policy function (what you want to

solve for, like $c(w, y)$) on some compact set by a polynomial. See Pohl et al. (2017) for a nice application of the projection method.

By Theorem 3 if you want to approximate a smooth function f on $[-1, 1]$ by a degree $N-1$ polynomial, it is "optimal" to interpolate f at the roots of the degree N Chebyshev polynomial. The idea of the projection method is to approximate the policy function f by a linear combination of Chebyshev polynomials,

$$f(x) \approx \widehat{f}(x) = \sum_{n=0}^{N-1} a_n T_n(x),$$

and determine the coefficients $\{a_n\}_{n=0}^{N-1}$ to make the functional equation (that you want to solve) true at the Chebyshev nodes.

It is easier to see how things work by looking at an example. Suppose you want to solve the differential equation

$$y'(t) = y(t),$$

with initial condition $y(0) = 1$. Of course the solution is $y(t) = \mathrm{e}^t$, but let's pretend that we don't know the solution and solve it numerically. Suppose we want to compute a numerical solution for $t \in [0, T]$. We can do as follows.

1. Map $[0, T]$ to $[-1, 1]$ by the affine transformation $t \mapsto \frac{2t-T}{T}$.
2. Approximate $y(t)$ by $\widehat{y}(t) = \sum_{n=0}^{N-1} a_n T_n(\frac{2t-T}{T})$
3. Determine $\{a_n\}_{n=0}^{N-1}$ by setting $\widehat{y}'(t) = \widehat{y}(t)$ at t corresponding to Chebyshev nodes for degree N (find t_n by solving $\frac{2t_n-T}{T} = \cos\left(\frac{2n-1}{2N}\pi\right)$ for $n = 1, \ldots, N$).
4. In this case, must also impose initial condition $\widehat{y}(0) = 1$, so for example can minimize sum of squared residuals at Chebyshev nodes:

$$\begin{aligned} &\underset{\{a_n\}_{n=0}^{N-1}}{\text{minimize}} && \sum_{n=0}^{N-1} (\widehat{y}'(t_n) - \widehat{y}(t_n))^2 \\ &\text{subject to} && \widehat{y}(0) = 1. \end{aligned}$$

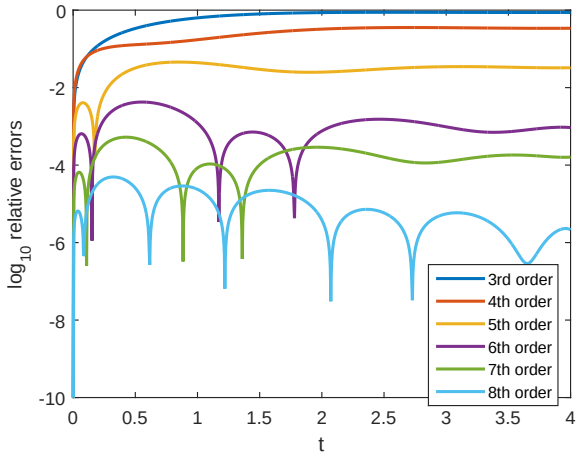
The figure below shows the $\log_{10}$ relative errors when $T = 4$ and $N - 1 = 3, 4, \ldots$. You can see that the relative errors become smaller as we increase the degree of polynomial approximation.

References

Jérôme Adda and Russel W. Cooper. *Dynamic Economics: Quantitative Methods and Applications*. MIT Press, Cambridge, MA, 2003.

Jonathan M. Borwein and Adrian S. Lewis. Duality relationships for entropy-like minimization problems. *SIAM Journal on Control and Optimization*, 29 (2):325–338, March 1991. doi:10.1137/0329017.

Philip J. Davis and Philip Rabinowitz. *Methods of Numerical Integration*. Academic Press, Orlando, FL, second edition, 1984.



Leland E. Farmer and Alexis Akira Toda. Discretizing nonlinear, non-Gaussian Markov processes with exact conditional moments. *Quantitative Economics*, 8(2):651–683, July 2017. doi:10.3982/QE737.

Kenneth L. Judd. Projection methods for solving aggregate growth models. *Journal of Economic Theory*, 58(2):410–452, December 1992. doi:10.1016/0022-0531(92)90061-L.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, March 1951. doi:10.1214/aoms/1177729694.

Walter Pohl, Karl Schmedders, and Ole Wilms. Higher-order effects in asset-pricing models with long-run risk. *Journal of Finance*, 2017. URL `https://ssrn.com/abstract=2540586`. Forthcoming.

K. Geert Rouwenhorst. Asset pricing implications of equilibrium business cycle models. In Thomas F. Cooley, editor, *Frontiers of Business Cycle Research*, chapter 10, pages 294–330. Princeton University Press, 1995.

Ken'ichiro Tanaka and Alexis Akira Toda. Discrete approximations of continuous distributions by maximum entropy. *Economics Letters*, 118(3):445–450, March 2013. doi:10.1016/j.econlet.2012.12.020.

Ken'ichiro Tanaka and Alexis Akira Toda. Discretizing distributions with exact moments: Error estimate and convergence analysis. *SIAM Journal on Numerical Analysis*, 53(5):2158–2177, 2015. doi:10.1137/140971269.

George Tauchen. Finite state Markov-chain approximations to univariate and vector autoregressions. *Economics Letters*, 20(2):177–181, 1986. doi:10.1016/0165-1765(86)90168-0.

George Tauchen and Robert Hussey. Quadrature-based methods for obtaining approximate solutions to nonlinear asset pricing models. *Econometrica*, 59 (2):371–396, March 1991. doi:10.2307/2938261.